

**Listing of Claims**

1. (Currently Amended) A computer-implemented method for dynamic emulation of legacy instructions of a legacy program where the legacy instructions are emulated a number of times comprising:
  - providing legacy state information for determining a program execution mode for emulating said legacy instructions each time the legacy instructions are emulated,
  - accessing said legacy instructions and said legacy state information each time the legacy instructions are emulated,
  - for each particular legacy instruction for each time the legacy instructions are emulated,
  - querying to determine if one or more particular translated instructions ~~for said execution mode~~ are stored as a result of translating said legacy instruction ~~for said execution mode,~~ and
    - if not translated ~~for said execution mode,~~
      - translating the particular legacy instruction into one or more particular translated instructions for emulating the particular legacy instruction for said execution mode,
    - storing said one or more particular translated instructions and storing with said legacy state information as stored state information with said one or more particular translated instructions, and
  - if translated ~~for said execution mode, continuing without additional translating,~~
    - querying said stored state information for said particular translated instructions and determining if a match occurs between the legacy state information and the stored state information,

if a match occurs, continuing without additional  
translating,

if no match occurs, performing said translating and  
storing steps,

accessing said one or more particular translated instructions for emulating said  
legacy instructions for said execution mode each time the legacy  
instructions are emulated,

executing said translated instructions to emulate said legacy instructions.

2. (Currently Amended) The method of Claim 1 wherein said storing of the one or more particular translated instructions is in one or more particular translated blocks and said stored state information is stored as a state word in each of said particular translated blocks.
3. (ORIGINAL) The method of Claim 1 wherein said legacy instructions are for a legacy system having a S/390 architecture.
4. (ORIGINAL) The method of Claim 1 wherein said legacy instructions are object code instructions compiled/assembled for a legacy architecture.
5. (ORIGINAL) The method of Claim 1 wherein said translated instructions are for execution in a RISC architecture.

6. (Currently Amended) A computer-implemented method for dynamic emulation of legacy instructions where the legacy instructions are emulated a number of times, where the legacy instructions are compiled/assembled into object code form for a native architecture, where the legacy instructions are executed as guests in the host architecture, where the legacy instructions are translated to translated instructions in the host architecture and the translated instructions are executed in the host architecture concurrently with the translation of the legacy instructions in the host architecture, comprising:

providing legacy state information for determining a program execution mode for emulating said legacy instructions each time the legacy instructions are executed,  
accessing said legacy instructions and said legacy state information as guests in the host architecture each time the legacy instructions are executed,

for each particular legacy instruction for each time the legacy instructions are executed,  
querying to determine if one or more particular translated instructions ~~for said execution mode~~ are stored as a result of translating said legacy instruction ~~for said execution mode~~, and

if not translated ~~for said execution mode~~,

translating the particular legacy instruction into one or more particular translated instructions for emulating the particular legacy instruction for said execution mode,

storing said one or more particular translated instructions ~~with~~ and storing said legacy state information as stored state information with said one or more particular translated instructions, and

~~if translated for said execution mode, continuing without additional translating,~~

querying said stored state information for said particular translated instructions and determining if a match occurs between the legacy state information and the stored state information,

if a match occurs, continuing without additional translating,

if no match occurs, performing said translating and storing steps,

accessing said one or more particular translated instructions for emulating said legacy instructions for said execution mode as a guest in said host architecture each time the legacy instructions are executed,

executing said translated instructions to emulate said legacy instructions.

7. (New) The method of Claim 6 wherein said storing of the one or more particular translated instructions is in one or more particular translated blocks and said stored state information is stored as a state word in each of said particular translated blocks.
8. (New) The method of Claim 7 wherein said storing of the one or more particular translated instructions is in a plurality of particular translated blocks and said stored state information is stored as a state word in each of said plurality of particular translated blocks.
9. (New) The method of Claim 6 wherein said storing of the one or more particular translated instructions is in one or more particular translated blocks and said stored state information is stored as a state word in each of said particular translated blocks and wherein a transfer routine is called at an end of each block to locate a next block.